

**PROGRAM MAINTENANCE MANUAL  
FOR THE GOES SATELLITE-DERIVED  
SEA SURFACE TEMPERATURE  
PROCESSING SYSTEM**

**December 8, 2000**

*Prepared by:*

Steve Olson  
National Oceanic and Atmospheric Administration  
National Environmental Satellite, Data, and  
Information Service  
Suitland, MD 20233

# Table Of Contents

<b>1. Program Maintenance Manual Overview .....</b>	<b>5</b>
<b>2. Introduction.....</b>	<b>6</b>
<b>3. Operating Environment .....</b>	<b>7</b>
<b>3.1. Hardware .....</b>	<b>7</b>
<b>3.2. Software .....</b>	<b>9</b>
<b>3.2.1 McIDAS-X .....</b>	<b>9</b>
<b>3.2.2 Abstract Data Distribution Environment (ADDE).....</b>	<b>9</b>
<b>3.2.3 Programming Language - VS FORTRAN Version 2 .....</b>	<b>9</b>
<b>3.2.4 GOES SST Processing Software .....</b>	<b>9</b>
<b>3.3. Operating Systems .....</b>	<b>10</b>
<b>3.3.1 IBM AIX Operating System .....</b>	<b>10</b>
<b>3.3.2 SGI IRIX Operating System .....</b>	<b>10</b>
<b>3.4. Data Transfer/Communications .....</b>	<b>10</b>
<b>3.4.1 File Transfer Protocol (FTP) .....</b>	<b>10</b>
<b>3.5. Support Software &amp; Interfaces .....</b>	<b>11</b>
<b>3.6 Security .....</b>	<b>11</b>
<b>4. Design Details - Process Descriptions .....</b>	<b>12</b>
<b>4.1 GOES SST Processing Overview .....</b>	<b>12</b>
<b>4.2 GOES SST Product Processing .....</b>	<b>14</b>
<b>4.2.1 Process 1: GVAR Data Ingest/McIDAS Decoding .....</b>	<b>17</b>
<b>4.2.1.1 Description .....</b>	<b>17</b>
<b>4.2.2 Process 2: Image Area File Transfer .....</b>	<b>18</b>
<b>4.2.2.1 Description .....</b>	<b>18</b>
<b>4.2.2.2 Program Names and Functions .....</b>	<b>18</b>
<b>4.2.3 Process 3: Creation of Multispectral AREA Files .....</b>	<b>18</b>
<b>4.2.3.1 Description .....</b>	<b>18</b>
<b>4.2.3.2 Program Names and Functions .....</b>	<b>18</b>
<b>4.2.4 Process 4: Creation of 1/2 hourly SST AREA Sector Files .....</b>	<b>19</b>
<b>4.2.4.1 Description .....</b>	<b>19</b>
<b>4.2.4.2 Program Names and Functions .....</b>	<b>19</b>
<b>4.2.5 Process 5: Creation of hourly composite and 3-hourly composite and sector SST AREA Files .....</b>	<b>19</b>
<b>4.2.5.1 Description .....</b>	<b>19</b>
<b>4.2.5.2 Program Names and Functions .....</b>	<b>19</b>
<b>4.2.6 Process 6: Collocation of SST Products with Buoy Measurements .....</b>	<b>20</b>
<b>4.2.6.1 Description .....</b>	<b>20</b>
<b>4.2.6.2 Program Names and Functions .....</b>	<b>20</b>
<b>4.2.7 Process 7: Product Distribution .....</b>	<b>20</b>
<b>4.2.7.1 Description .....</b>	<b>20</b>
<b>4.2.7.2 Program Names and Functions .....</b>	<b>20</b>
<b>5. Detailed Software Descriptions .....</b>	<b>22</b>
<b>5.1 UNIX Shell Scripts .....</b>	<b>22</b>
<b>5.1.1 3HOURLY.KSH .....</b>	<b>22</b>

<b>5.1.2</b>	<b>BUOYMATCH.KSH .....</b>	<b>22</b>
<b>5.1.3</b>	<b>CLEAN2000.KSH .....</b>	<b>23</b>
<b>5.1.4</b>	<b>CLEANUP.KSH.....</b>	<b>23</b>
<b>5.1.5</b>	<b>GETBUOY.SCR.....</b>	<b>23</b>
<b>5.1.6</b>	<b>GETDATA_HH15.KSH.....</b>	<b>24</b>
<b>5.1.7</b>	<b>GETDATA_HH45.KSH.....</b>	<b>24</b>
<b>5.1.8</b>	<b>HOURLYSST.KSH .....</b>	<b>25</b>
<b>5.1.9</b>	<b>PROCGSST_HH15.KSH .....</b>	<b>25</b>
<b>5.1.10</b>	<b>PROCGSST_HH45.KSH .....</b>	<b>27</b>
<b>5.2</b>	<b>Main Programs .....</b>	<b>29</b>
<b>5.2.1</b>	<b>AVGIMI.PGM (Spatial Reduction Program) .....</b>	<b>29</b>
<b>5.2.2</b>	<b>BUOY.PGM (Co-locate program) .....</b>	<b>29</b>
<b>5.2.3</b>	<b>GIMIGLUE.C (Band Merging Program) .....</b>	<b>30</b>
<b>5.2.4</b>	<b>GOES_SST.PGM (Main SST Processing Program) .....</b>	<b>31</b>
<b>5.2.5</b>	<b>MERGE.PGM (Hourly and 3-hourly composite SST program) .....</b>	<b>32</b>
<b>5.3</b>	<b>Subprogram Descriptions .....</b>	<b>32</b>
<b>5.3.1</b>	<b>AVGIMI.PGM Subprogram Descriptions .....</b>	<b>32</b>
5.3.1.1	ARAGET (AREA,OFFSET,NBYTE,OUTBUF) .....	32
5.3.1.2	ARAOPT(AREA,NOPT,COPT,VAL) .....	33
5.3.1.3	ARAPUT(AREA,OFFSET,NBYTE,IARRAY) .....	33
5.3.1.4	CFI(NUMB) .....	33
5.3.1.5	CLSARA(AREA) .....	34
5.3.1.6	HOWBIG(AREA,NLIN,NELE) .....	34
5.3.1.7	OPNARA(AREA) .....	34
5.3.1.8	READD(ANUM,ENTRY) .....	35
5.3.1.9	REDARA(AREA,LINE,ELEM,NELE,BAND,IARRAY) .....	35
5.3.1.10	WRTARA(AREA,LINE,IARRAY) .....	35
5.3.1.11	ZEROS(BUF,BYTES) .....	36
5.3.1.12	ZEROW(NWORDS,IARR) .....	36
<b>5.3.2</b>	<b>BUOY.PGM Subprogram Descriptions .....</b>	<b>36</b>
5.3.2.1	ARAOPT(AREA,NOPT,COPT,VAL) .....	36
5.3.2.2	CLSARA(AREA) .....	37
5.3.2.3	NO_GOES(IMAGE, REF_SEC, LAT, LON, SAT_ID, SAT_REF, CLEAR, ANGLES, BUF) .....	37
5.3.2.4	NV1EAS(ZLAT,ZLON,XLIN,XELE,XDUM) .....	37
5.3.2.5	NV1OPT(IFUNC,XIN,XOUT) .....	38
5.3.2.6	NVSET(COPT,NUM) .....	38
5.3.2.7	OPNARA(ANUM,ENTRY) .....	38
5.3.2.8	READD(ANUM,ENTRY) .....	39
5.3.2.9	REDARA(AREA,LINE,ELEM,NELE,BAND,IARRAY) .....	39
<b>5.3.3</b>	<b>GOES_SST.PGM Subprogram Descriptions .....</b>	<b>39</b>
5.3.3.1	ARAOPT(AREA,NOPT,COPT,VAL) .....	39
5.3.3.2	CLSARA(AREA) .....	40
5.3.3.3	GOES_SST(IMAGE) .....	40
5.3.3.4	NAV_PIX(IMAGE,TWILIGHT,LINE,ELEM,SCENE,SAT_Z,SUN_Z) .....	40
5.3.3.5	OPNARA(AREA) .....	41
5.3.3.6	READ_AREA(AREA,LINE,B_ELEM,N_ELEM,BAND,BUF,VAL_LO,VAL_HI,MIS_LO,MIS_HI) .....	41
5.3.3.7	REDARA(AREA,LINE,ELEM,NELE,BAND,IARRAY) .....	41

5.3.3.8 STATS(X,Y,N,JS,JE,IS,IE,MEAN,VARI) .....	42
5.3.3.9 WRTARA(AREA,LINE,IARRAY) .....	42
<b>5.3.4 MERGE.PGM Subprogram Descriptions .....</b>	<b>43</b>
5.3.4.1 ARAOPT(AREA,NOPT,COPT,VAL) .....	43
5.3.4.2 CLSARA(AREA) .....	43
5.3.4.3 NV1EAS(ZLAT,ZLON,XLIN,XELE,XDUM) .....	43
5.3.4.4 NVSET(COPT,NUM) .....	44
5.3.4.5 OPNARA(AREA) .....	44
5.3.4.6 READD(ANUM,ENTRY) .....	44
5.3.4.7 REDARA(AREA,LINE,ELEM,NELE,BAND,IARRAY) .....	45
<b>6. Database Specification.....</b>	<b>46</b>
<b>6.1 Input File Specifications .....</b>	<b>46</b>
6.1.1 Calibration Coefficient files .....	46
<b>6.2 Output File Specifications .....</b>	<b>46</b>
6.2.1 GETDATA_HH15.LOG .....	46
6.2.2 PROCGSST_HH15.LOG .....	48
6.2.3 GETDATA_HH45.LOG .....	50
6.2.4 PROCGSST_HH45.LOG .....	52
<b>7. Installation Procedures .....</b>	<b>54</b>
7.1 Installation of Software and Data .....	54
7.2 Installation Changes Concerning System/Configuration Upgrades .....	55
7.2.1 McIDAS-X Upgrades .....	55
<b>8. Utilities For System Observation And Data Analysis .....</b>	<b>56</b>
8.1 Utilities For System Observation/Maintenance/Analysis .....	56
8.1.1 Utility Listing .....	56
8.1.2 Utility Examples .....	56
<b>9. References.....</b>	<b>57</b>

# 1. Program Maintenance Manual Overview

This ***Program Maintenance Manual*** provides a comprehensive description of the GOES Sea Surface Temperature (SST) software processing system. This includes a description of each processing element of the system and the software involved in each element. It is intended to be thorough, allowing for a complete understanding of the system so that it can be satisfactorily maintained. Each chapter in this document gives the reader a different level of understanding into the GOES SST processing system. The first two chapters outline the basic elements of the system. They introduce the hardware, operating environment, and a high level introduction to the processes involved. Chapter 3 provides a detailed description of each process comprising the SST processing system. Chapters 4 and 6 detail the internals of the system which include the algorithms, software, and files. Chapters 5 and chapters 7-9 describe the installation and maintenance procedures for this system. Below is a brief summary of each chapter.

<b>Introduction:</b>	This chapter is an overview of the entire SST processing system operating in the workstation environment. It contains only brief descriptions of data transfers and processes, and does not discuss procedures in detail.
<b>Operating Environment:</b>	This chapter describes the setup and environment in which the SST production process resides.
<b>Design Details - Process Descriptions:</b>	This chapter discusses each major process involved in the production of SST products. There is a description of each major process, program names, file information, diagnostic output, and procedures for monitoring, troubleshooting, and recovery.
<b>Program Documentation:</b>	This chapter defines the main programs, subroutines, and functions used in the processing.
<b>Database Specifications:</b>	This chapter describes the formats of the various files used in the SST processing system.
<b>Installation Procedures:</b>	This chapter outlines detailed procedures for installing and configuring the GOES SST product system. This includes installing and compiling the SST software and scheduling jobs on the UNIX workstation.
<b>System Maintenance:</b>	This chapter discusses possible system problems and the steps to take to reestablish McIDAS-X software and reboot the UNIX workstation.
<b>Utilities:</b>	This chapter provides examples of numerous utility programs to aid the system analyst in looking at the data which may aid in the diagnosing of system problems.
<b>References:</b>	This chapter provides a listing of relevant scientific articles on satellite-derived SST and McIDAS references.

## **2. Introduction**

The GOES Satellite Derived SST Product System is designed to incorporate GOES multispectral information for channels 1,2,4 and 5 to derive sea-surface temperature. Extensive screening for cloud contamination affect the regression equation are performed on these channel measurements to yield clear sky radiance values. Four regression equations are produced for GOES-E and GOES-W during the day and night covering 30°W - 180°W, 45°S - 60°N.

$$SST = C_0 + C_1 T_4 + C_2 dT + C_3 dT^2 + C_4 (\sec \alpha - 1)$$

where,

$C_i$ ,  $i=1,4$  are provided in regression coefficients and may be updated from time to time,  $dT = T_{11} - T_{12}$  for daytime only and  $T_{3,9} - T_{11}$  for nighttime only, and  $\alpha$  is the satellite viewing angle.

Production of the experimental GOES SST began on October 1, 1998. The current GOES operational SST production system runs in an automated fashion every half hour. Products from the SST Processing System include hourly composite SST, 3-hour composite SST, and sector SST imagery for CoastWatch regions for the NorthEast, SouthEast, Great Lakes, Gulf of Mexico, La Hoya, Hawaii and Fairbanks, Alaska Node Sites.

### **3. Operating Environment**

The operating environment for the GOES SST processing system is shown in the hardware configuration which is depicted in Figure 3-1.

#### **3.1. Hardware**

The hardware configuration utilized by the GOES SST Processing System consists of Silicon Graphics Origin 2000 workstations and International Business Machines (IBM) RS-6000 UNIX workstations.

The GOES VARiable (GVAR) data is ingested on a Silicon Graphics workstation and decoded into the proprietary McIDAS area file format and staged on two different IBM SP workstations . These two workstations serve as the primary GOESNET servers for GOES-8 (GER server) and GOES-10 (GWR server) imagery . The satellite-derived SST product system runs on an SGI Origin 2000 workstation and accesses Buoy data from the IBM SP's. The final satellite-derived SST products are kept on the SGI Origin 2000 workstation for a 24 hour period and subsequently distributed to an IBM RS-6000 Model 590 which serves as the SATEPS product server (DUC6).

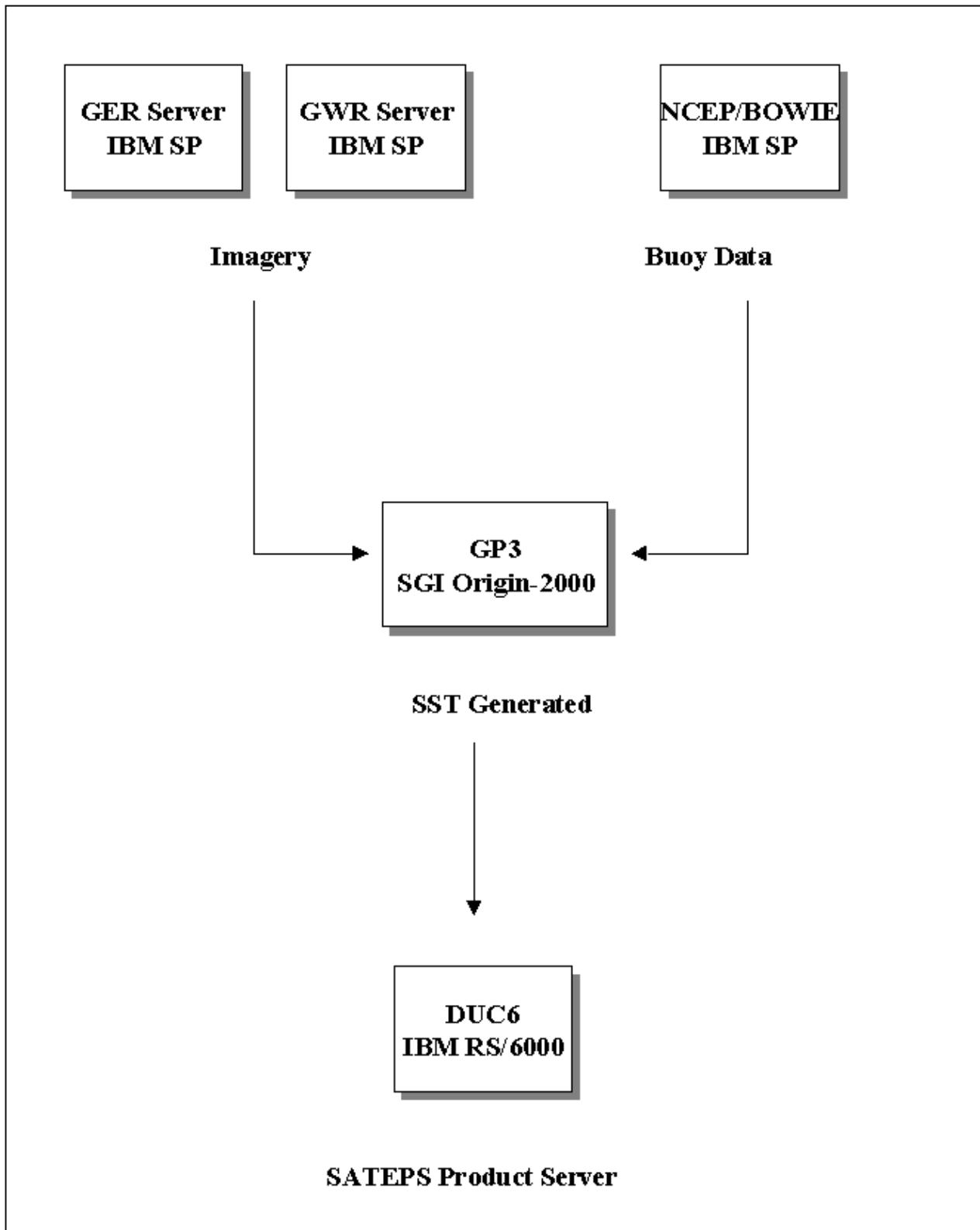


Figure 3-1. Hardware configuration for the GOES Satellite derived SST Processing System

## **3.2. Software**

### **3.2.1 McIDAS-X**

The Man computer Interactive Data Access System (McIDAS) is a software package developed, maintained, and supported by the Space Science and Engineering Center (SSEC). The McIDAS-X software package is designed to run on a variety of powerful UNIX workstations. Functionality provided by McIDAS-X includes data management and data analysis which can support both meteorological operations and research. The McIDAS database contains both real-time weather satellite data and conventional weather data. For more information regarding McIDAS-X, the reader is referred to the McIDAS-X Users Guide ([http://www.ssec.wisc.edu/mug/users\\_guide/USERS\\_GUIDETOP.html](http://www.ssec.wisc.edu/mug/users_guide/USERS_GUIDETOP.html)) and the McIDAS-X Learning Guide ([http://www.ssec.wisc.edu/mug/learn\\_guide/LEARN\\_GUIDETOP.html](http://www.ssec.wisc.edu/mug/learn_guide/LEARN_GUIDETOP.html)).

### **3.2.2 Abstract Data Distribution Environment (ADDE)**

The Abstract Data Distribution Environment (ADDE) software package was developed by SSEC which uses clients and servers in a distributed data environment. Servers store data and distribute it to the client. Clients and servers communicate via the Transmission Control Protocol/Internet Protocol (TCP/IP). The ADDE design is intended to exploit the CPU cycles on a client workstation and allow easy access to data on multiple servers across local and wide area networks. Servers store data and pass it to the client. Clients request and receive data, and run applications on the data.

McIDAS-X workstations can be configured to act as both a client or local server. The client can request data from the local server or a remote server, with the remote server being a different account on the same McIDAS-X workstation configured as a remote server, an account on another McIDAS-X workstation configured as a remote server, or a McIDAS-MVS mainframe configured as a server. The GOES SST Processing System uses the ADDE software package to locate and transfer McIDAS Area files from the IBM SP's (server) to the IBM RS/6000 workstation (client). For more information regarding the ADDE software the reader is referred to the McIDAS-X Users Guide ([http://www.ssec.wisc.edu/mug/users\\_guide/USERS\\_GUIDETOP.html](http://www.ssec.wisc.edu/mug/users_guide/USERS_GUIDETOP.html)).

### **3.2.3 Programming Language - VS FORTRAN Version 2**

The VS FORTRAN Version 2 language is best suited to applications that involve mathematical computations and other manipulations of arithmetic data. The language consists of a set of character conventions and rules that are used to convey information to the compiler. The basis of the language is a statement containing combinations of names, operators, constants, and words (keyword) whose meaning is predefined to the compiler.

### **3.2.4 GOES SST Processing Software**

All of the software for the GOES SST processing software will reside on the SGI Origin 2000 workstation under a single directory. This operating system is currently in place and running at NOAA/NESDIS Satellite Data Distribution System. To view products generated from this system, please visit <http://www.ssd.noaa.gov/SSD/ML/prodsvcs.html>.

### **3.3. Operating Systems**

The hardware utilized by the GOES SST processing system consists of SGI Origin 2000 and IBM RS/6000 Model 590/591 UNIX workstations.

#### **3.3.1 IBM AIX Operating System**

The Advanced Interactive Executive (AIX) operating system is the IBM variant of UNIX. AIX Version 3.2.5 is the current base operating system version installed on the DUC workstations.

#### **3.3.2 SGI IRIX Operating System**

The IRIX operating system is the SGI variant of UNIX. IRIX Version 6.5 is the current base operating system version installed on the SGI Origin 2000 workstations.

### **3.4. Data Transfer/Communications**

The GOES SST Processing System uses the ADDE software package to transfer McIDAS Area files from the GOES servers (GER,GWR) to the SGI Origin 2000 workstation (client). Clients and servers communicate via the Transmission Control Protocol/Internet Protocol (TCP/IP).

#### **3.4.1 File Transfer Protocol (FTP)**

The File Transfer Protocol (FTP) is a part of the Transmission Control Protocol/Internet Protocol (TCP/IP). The file mechanism used to transfer files from the host to the foreign data center is the TCP/IP protocol, using the Ethernet communications network. The FTP allows the transfer of data between local and foreign hosts or between two hosts. FTP is built on the services of TCP in the transport layers. FTP transfers data sets as either ASCII characters or as binary data. ASCII characters are used to transfer data sets that contain only text characters.

Using the FTP command and its subcommands, multiple foreign hosts can be sequentially accessed without leaving the FTP environment. Specifically, FTP allows the user to:

- Establish a connection to a foreign host's FTP server
- Identify the user to the foreign host's FTP server
- Obtain status information about FTP on the foreign host
- List directories belonging to the foreign host
- Transfer datasets to and from the foreign host
- Delete or rename datasets on the foreign host
- Enter TSO commands
- Obtain assistance for the FTP sub-commands
- Input from or write to a dataset or file
- Exit the FTP environments when an error occurs

Security is handled by passing user and account passwords to the foreign hosts.

TCP provides a reliable vehicle for delivering packets between hosts on an internet. TCP takes a stream of data, breaks it into datagrams, sends each one individually using IP, and reassembles the datagrams at the destination node. If any datagrams are lost or damaged during transmission, TCP detects this and resends the missing datagrams. The received data stream is a reliable copy of the transmitted data stream.

IP provides the interface from the transport layer (host-to-host) protocols to the physical-level protocols. IP is the basic transport mechanism for routine IP packets to the next gateway, router, or destination host.

IP provides the means to transmit blocks of data (or packets of bits) from sources to destination. Destinations are hosts identified by fixed-length addresses. Outgoing packets automatically have an IP header prefixed, and incoming packets have their IP header removed before being sent to the higher-level protocols. This protocol provides for the universal addressing hosts in an internet network.

IP does not ensure reliable communication, because it does not require acknowledgements from the sending hosts, the receiving host, or intermediate hosts. IP does not provide error control for the data; it provides only a header checksum. IP treats each packet as an independent entity unrelated to any other packet. IP does not perform retransmissions or flow control. A higher-level protocol that uses IP must implement its own reliability procedures.

### **3.5. Support Software & Interfaces**

Support software consists primarily of core McIDAS-X routines. Some primary software groups utilized are jobs to control batch commands, scripts, scheduling, and commands which do simple operations (display, list, etc.) on McIDAS-specific files (area files, MD files, grid files, and LW files).

### **3.6 Security**

The GOES SST processing software and data on the SGI Origin 2000 workstation resides in directories under restricted user accounts. Only certain individuals in operations have access to these accounts. Frequent updating of user passwords for these accounts also maintains the integrity of the software and data residing in these accounts.

Software revision control will be maintained via the Revision Control System (RCS) software. Use of this software package, in conjunction with frequent system tape backups, will ensure the integrity of the operational SST software and data.

## **4. Design Details - Process Descriptions**

### **4.1 GOES SST Processing Overview**

An overview of the entire GOES Satellite Derived SST Processing System is shown in Figure 4-1. As is evident from this figure, there are numerous processes involved in the production of the satellite-derived SST products. Each process shown in Figure 4-1 is numbered and will be described in the following sections. The processing segments may be categorized into three areas: SST pre-processing, SST processing, and SST post-processing. For those processes within the scope of the GOES SST Processing System, the following information is included:

- Description of each major process
- Names of each program, macro, batch file, and script used in each process
- Description of the initiation method of each process and a listing of scheduled processes
- Information for files (input and output) created or updated
- Listing or partial listing of diagnostic output generated

For those processes beyond the scope of this document, a brief description of the process will be provided which will include a discussion of the relevant data involved in the process. This information was added for the sake of completeness.

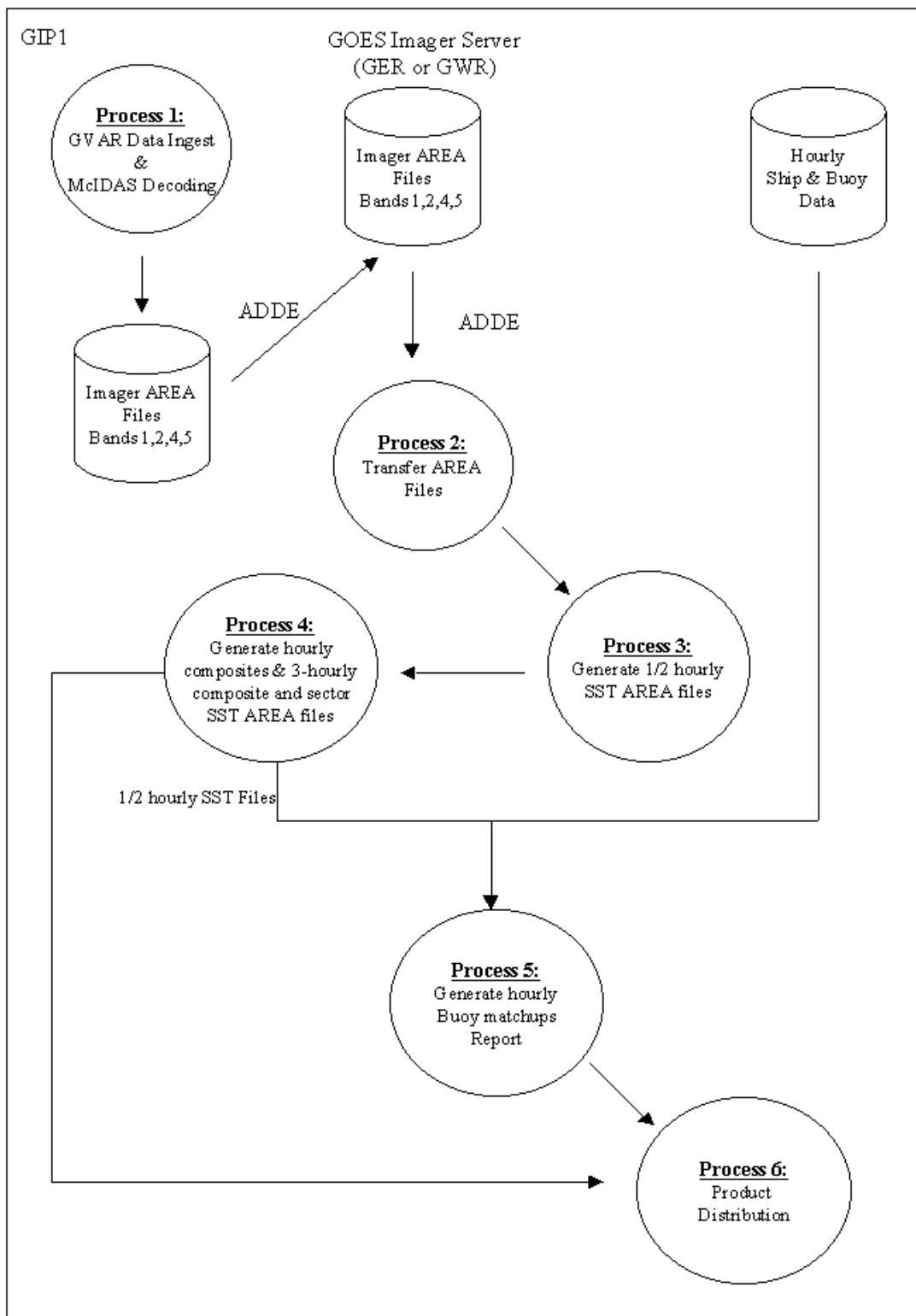


Figure 4-1. The GOES Satellite Derived SST Processing System

## **4.2 GOES SST Product Processing**

The numerous processing steps involved in the production of GOES satellite-derived SST is contained and controlled within several processing scripts. All of the scripts are controlled by a single cron job. The cron job is used to schedule the time of day to invoke each of the scripts. Figure 4-2 indicates when each script is invoked. While it takes several scripts to generate the satellite-derived SST products, figure 4-3 describes the whole process through a flow diagram. The various processing steps, and the programs which make up these processes, are shown in this figure and described in detail in the following subsections, 4.2.1-4.2.8.

## GOES SST PROCESSING SCRIPTS

SCRIPT	MINUTE	HOUR	DAY	MONTH	DAY OF WEEK
getdata_hh15.ksh	05	*	*	*	*
procgsst_hh15.ksh	25	*	*	*	*
getdata_hh45.ksh	35	*	*	*	*
procgsst_hh45.ksh	55	*	*	*	*
cleanup.ksh	05	00	*	*	*
clean2000.ksh	15	00	*	*	*
hourlysst.ksh	05	*	*	*	*
3hourly.ksh	05	00,03,06,09, 12,15,18,21	*	*	*
buoymatch.ksh	40	*	*	*	*
getbuoy.ksh	55	*	*	*	*

Figure 4-2     SST Processing scripts and when each is run. A '\*' indicates that it is to run continuously for that given parameter

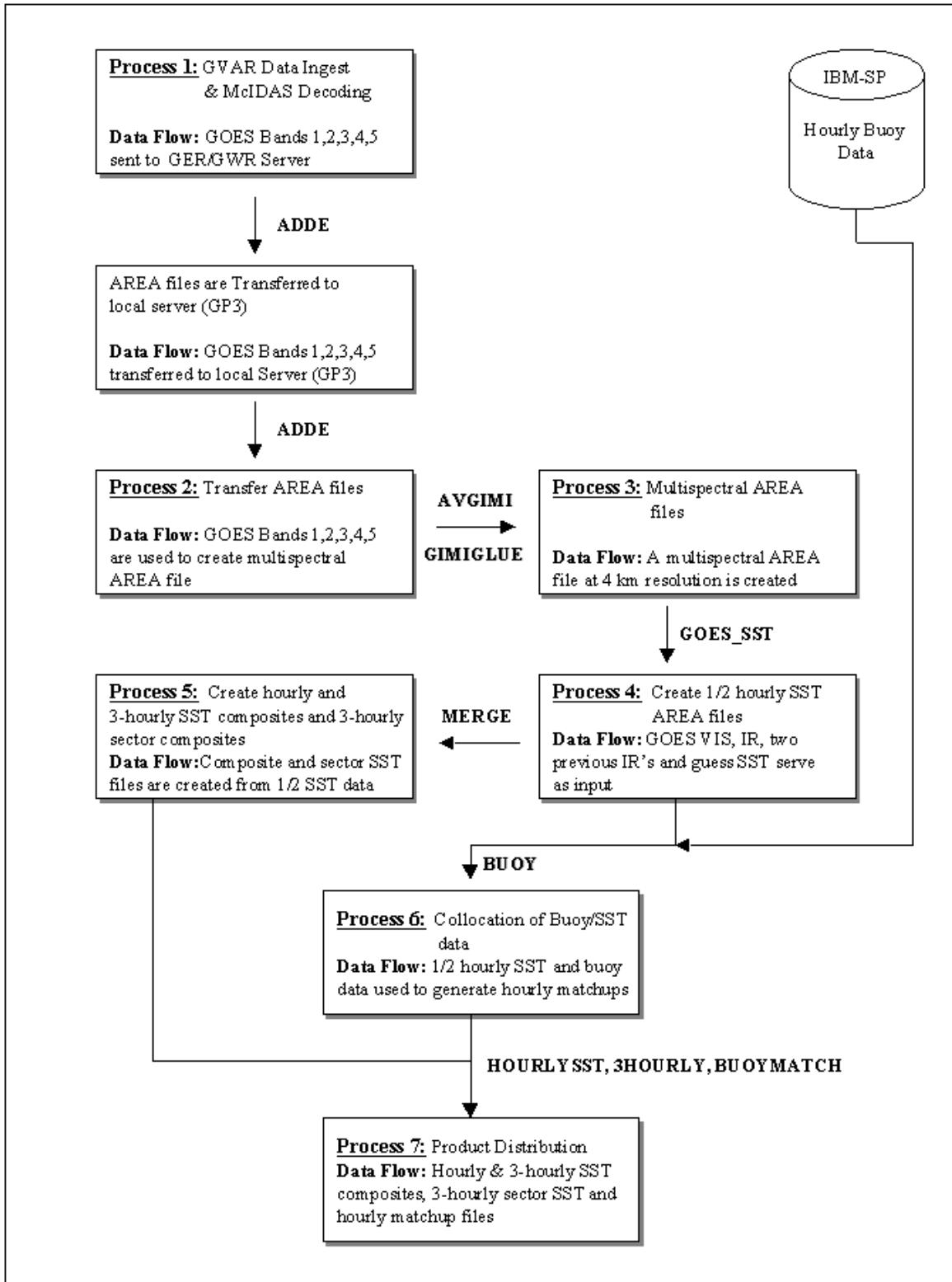


Figure 4-3. Data flow diagram of the GOES Satellite Derived SST Processsing System

## **4.2.1 Process 1: GVAR Data Ingest/McIDAS Decoding**

### **4.2.1.1 Description**

The GOES SST processing system is dependent upon a successful GVAR ingest and decoding process. The ingesting of GVAR data and the subsequent decoding of this data into McIDAS imager area files provides the satellite imagery necessary for the production of GOES SST products. The GOES imager channels 1-5 are decoded into separate area files on the GOES ingestor. For imager-based SST processing, all 5 bands are transferred to the appropriate server (GER for GOES-EAST; GWR for GOES-WEST). ADDE software is used to transfer the imager areas from the GOES image servers to the SGI Origin 2000 SST workstation. The ADDE dataset names, for imagery, used by the SST processing system are given in Table 4-1:

<b><i>GOES-EAST Server (GER) ADDE Group/Dataset Name</i></b>	<b><i>Description</i></b>	<b><i>GOES-WEST Server (GWR) ADDE Group/Dataset Name</i></b>
GER/GENHEM01V	Imager Band 1; extended Northern Hemisphere (1 km res)	GWR/GWNHEM01V
GER/GENHEM04I2	Imager Band 2; extended Northern Hemisphere (4 km res)	GWR/GWNHEM01I2
GER/GENHEM04I4	Imager Band 4; extended Northern Hemisphere (4 km res)	GWR/GWNHEM04I4
GER/GENHEM04I5	Imager Band 5; extended Northern Hemisphere (4 km res)	GWR/GWNHEM04I5
GER/GESHEM01V	Imager Band 1; extended Southern Hemisphere (1 km res)	GWR/GWSHEM01V
GER/GESHEM04I2	Imager Band 2; extended Southern Hemisphere (4 km res)	GWR/GWSHEM04I2
GER/GESHEM04I4	Imager Band 4; extended Southern Hemisphere (4 km res)	GWR/GWSHEM04I4
GER/GESHEM04I5	Imager Band 5; extended Southern Hemisphere (4 km res)	GWR/GWSHEM04I5

*Table 4-1 ADDE group and dataset names for GOES-EAST imagery residing on GER server and GOES-West imager residing on the GWR server.*

## ***4.2.2 Process 2: Image Area File Transfer***

### **4.2.2.1 Description**

This process involves the transfer of valid, high quality, GOES imagery for SST processing to the local server. An ADDE program **imgcopy** is used to transfer incoming GOES channel data to the local server. There are several criteria which must be met for accessing imagery for SST processing, including:

- 1) The time of each image is computed in each imgcopy cycle. Times are generally 30 minutes apart.

### **4.2.2.2 Program Names and Functions**

#### **a) IMGCOPY**

The ADDE program, IMGCOPY, is called from the getdata\_hh15.ksh, procgsst\_hh15.ksh, getdata\_hh145.ksh, procgsst\_hh45.ksh scripts and is used to transfer imagery from the appropriate GOES satellite data server (GER, GWR).

## ***4.2.3 Process 3: Creation of Multispectral AREA Files***

### **4.2.3.1 Description**

In order to generate SST products, the main program, GOES\_SST.PGM, requires multispectral AREA files containing GOES channels 1,2,4, and 5. Two programs (AVGIMI and GIMIGLUE) are used to generate the GOES multispectral datasets.

### **4.2.3.2 Program Names and Functions**

#### **a) AVGIMI**

AVGIMI.PGM performs pixel averaging to decrease image resolution. It is applied to the GOES Imager visible channel data to reduce it to the same resolution as the IR bands. AVGIMI differs from core McIDAS program AVGI; AVGIMI.PGM averages raw counts, while AVGI averages calibrated radiances.

#### **b) GIMIGLUE**

The program, gimiglue.c, reads in the single-band imager area files and creates the multispectral imager area file. This imager area file is the input to the SST product generation program.

#### **4.2.4 Process 4: Creation of 1/2 hourly SST AREA Sector Files**

##### **4.2.4.1 Description**

Multispectral GOES AREA files are used by the program, goes\_sst.k, to generate SST AREA sector files. During processing of SST AREA files, several quality control checks are performed, including:

Visible Reflectance > 0.1 and <=0.999  
IR brightness Temp (chns 3,4,5) > 150K and < 320K

##### **4.2.4.2 Program Names and Functions**

###### **a) GOES\_SST**

The main program, GOES\_SST.PGM, applies a regression-based algorithm to GOES multispectral satellite data to generate sector SST AREA files. The regression algorithm is based on buoy collocation data since November 1999. The algorithms for GOES-8 and GOES-10 are as follows:

$$SST = C_0 + C_1 T_4 + C_2 dT + C_3 dT^2 + C_4 (\sec q - 1)$$

where,

$C_i$ ,  $i=1,4$  are provided in regression coefficients defined in the data file reg\_coef.asc and may be updated from time to time,  $dT = T_{11} - T_{12}$  for daytime only and  $T_{3,9} - T_{11}$  for nighttime only, and  $q$  is the satellite viewing angle.

#### **4.2.5 Process 5: Creation of hourly composite and 3-hourly composite and sector SST AREA Files**

##### **4.2.5.1 Description**

This part of process uses the main program, MERGE.PGM, to take SST AREA files by sector for each half hour and create hourly composite and sector SST AREA files. Every third hour, the main program, MERGE.PGM, will also create 3-hourly composite SST AREA files.

##### **4.2.5.2 Program Names and Functions**

###### **a) MERGE**

The main program, MERGE.PGM, averages three SST fields within one hour to form a composite image. This fills many holes created by clouds. To remove stripping effects, two adjacent lines are averaged. The output from MERGE.PGM is either an hourly or 3-hourly composite SST field.

## ***4.2.6 Process 6: Collocation of SST Products with Buoy Measurements***

### **4.2.6.1 Description**

Comparisons of satellite derived SST to buoy data is done as a quantitative measure of the satellites quality. The statistics generated then comprise a baseline for comparison to evaluate changes that are introduced in an effort to improve the quality of the SST products. The main program, BUOY.PGM, is used to generate these comparisons of buoy and SST data.

### **4.2.6.2 Program Names and Functions**

#### **a) BUOY**

The main program, BUOY.PGM, is used to co-locate a buoy report with GOES measurements. Hourly buoy data are available 50 minutes after the hour in buoy.hh, where hh is hour. Hourly GOES imager data and GOES SST are available two hours after the hour. The SST data are used to indicate clear/cloudy pixels. Temporal co-location is done where the time difference among GOES, buoy, and the nominal hour is less than 30 minutes. Spatial co-location is done with standard McIDAS navigation. Results for each hour is written to a file yyyy\_ddd\_hh.

where,

yyyy = The four digit representation of the year

ddd = The julian date within a calendar year

hh = The two digit representation of hour

## ***4.2.7 Process 7: Product Distribution***

### **4.2.7.1 Description**

The UNIX shell scripts hourlysst.ksh, 3hourly.ksh, and buoymatch.ksh transfer the hourly, 3-hourly and buoy match data respectively from the mcidas data directory to the local archive directory. In doing so, the files are given more mnemonic names. The scripts then transfer the datasets, under the new names, to the SATEPS product server using ftp.

### **4.2.7.2 Program Names and Functions**

#### **a) HOURLYSST.KSH**

A UNIX korn-shell script to transfer the hourly SST files from the McIDAS data directory to the local archive directory.

**b) 3HOURLY.KSH**

A UNIX korn-shell script to transfer the 3-hourly SST and CoastWatch files from the McIDAS data directory to the local archive directory.

**c) BUOYMATCH.KSH**

A UNIX korn-shell script to transfer the hourly buoy matchup results from the McIDAS data directory to the local archive directory.

## 5. Detailed Software Descriptions

The following section contains detailed descriptions of individual pieces of software. All GOES SST scripts, main programs, subroutines, and functions (**non-core software**) will be detailed in this section.

For consistency and ease of comprehending the order of the software descriptions to follow will be: scripts, programs, and subprograms. The software descriptions for each software element will be arranged alphabetically.

### 5.1 UNIX Shell Scripts

This section provides detailed descriptions of script software elements in the SST processing system.

#### 5.1.1 3HOURLY.KSH

**Description** A UNIX Korn shell script to copy the 3-hourly SST and Coast Watch sector files from the McIDAS data directory to the local archive directory and to transfer via ftp to the product server via ftp.

**Input** No input required

**Design Details** The script performs several functions, including:

1. 3-Hourly SST files in the data directory (\$HOME/mcidas/data) are copied to the /data/sstarc directory for the 3-hour composite and all CoastWatch sectors.
2. The McIDAS procedure, ftp3hr.k, is used to transfer the SST files to the local archive directory.

**Keywords** None

#### 5.1.2 BUOYMATCH.KSH

**Description** A UNIX Korn shell script to copy the hourly buoy matchup results from the McIDAS data directory to the local archive directory and to transfer via ftp to the product server.

**Input** No input required

**Design Details** The script performs several functions, including:

1. Hourly buoy matchup results file in the data directory (/usr/people/goessst/mcidas/data) are copied to the /data/sstarc directory.

Infile file = \${year}\_\${day}.\${hour}  
Output file = match1\_\${year}\_\${day}\_\${hour}

where,

year = The four digit representation of the year

day = The three digit representation of julian day (Range is 0-365 in non-leap years and 0-366 in leap years.)

hour = The two digit representation of hour (Range is 0-23)

2. The core McIDAS procedure, ftpduc6.k, is used to transfer the hourly buoy matchup results to the local archive directory

**Keywords** None

### 5.1.3 CLEAN2000.KSH

**Description** A UNIX Korn shell script run nightly to clean up \$HOME/mcidas/data directory.

The script deletes N-day old SST output files (i.e. 2000\_243.17 and 2000\_244\_33A).

**Input** No input required

**Design Details** The script includes:

1. Change directory to \$HOME/mcidas/data
2. Remove N-day old files from the \$HOME/mcidas/data directory

**Keywords** None

### 5.1.4 CLEANUP.KSH

**Description** A UNIX Korn shell script to delete old SST output files from the archive directory.

**Input** No input required

**Design Details** The script includes:

1. Remove files from the /data/sstarc directory

```
rm /data/sst/*_${year}_${day}*
rm /data/sstarc/${year}_${day}**
```

**Keywords** None

### 5.1.5 GETBUOY.SCR

**Description** A UNIX Korn shell script to retrieve buoy data from the IBM\_SP via ftp.

**Input** No input required

**Design Details** The script includes:

1. Copies the file from ncep via ftp and renames the local copy to buoysst.hh, where hh is the two digit representation of hour.

**Keywords** None

### **5.1.6 GETDATA\_HH15.KSH**

**Description** A UNIX Korn shell script to transfer AREA files for three of the four sectors to local server for SST processing.

**Input** No input required

**Design Details** The script performs several functions, including:

1. Change directories to the McIDAS data directory (\$HOME/mcidas/data)
2. Acquire the data one sector at a time.

**A. GOES EAST Northern Hemisphere**

Script calls core McIDAS command imgdel to delete areas 1950 and 951-955

Script calls core McIDAS command imgcopy to copy data from GOES server to local server in McIDAS areas 1950 and 952-955.

**B. GOES EAST Southern Hemisphere**

Script calls core McIDAS command imgdel to delete areas 1970 and 971-975

Script calls core McIDAS command imgcopy to copy data from GOES server to local server in McIDAS areas 1970 and 972-975.

**C. GOES WEST Southern Hemisphere**

Script calls core McIDAS command imgdel to delete areas 1980 and 981-985

Script calls core McIDAS command imgcopy to copy data from GOES server to local server in McIDAS areas 1980 and 982-985.

**Keywords** None

### **5.1.7 GETDATA\_HH45.KSH**

**Description** A UNIX Korn shell script to transfer AREA files for three of the four sectors(NATL,SATL,SPAC) to local server for SST processing.

**Input** No input required

**Design Details** The script performs several functions, including:

1. Change directories to the McIDAS data directory (\$HOME/mcidas/data)
2. Acquire the data one sector at a time.

**A. GOES EAST Northern Hemisphere**

Script calls core McIDAS command imgdel to delete areas 1910 and 911-915

Script calls core McIDAS command imgcopy to copy data from GOES server to local server in McIDAS areas 1910 and 912-915.

**B. GOES EAST Southern Hemisphere**

Script calls core McIDAS command imgdel to delete areas 1930 and 931-935

Script calls core McIDAS command imgcopy to copy data from GOES server to local server in McIDAS areas 1930 and 932-935.

**C. GOES WEST Southern Hemisphere**

Script calls core McIDAS command imgdel to delete areas 1940 and 941-945

Script calls core McIDAS command imgcopy to copy data from GOES server to local server in McIDAS areas 1940 and 942-945.

**Keywords** None

### **5.1.8 HOURLYSST.KSH**

**Description** A UNIX Korn shell script to copy the hourly SST files from the McIDAS data directory to the local archive directory and transfer via ftp to the product server.

**Input** No input required

**Design Details** The script performs several functions, including:

1. Hourly SST files in the data directory (\$HOME/mcidas/data) are copied to the /data/sstarc/sst1 directory.
2. The core McIDAS procedure, ftpduc6.k, is used to transfer the SST files to the local archive directory

**Keywords** None

### **5.1.9 PROCGSST\_HH15.KSH**

**Description** A UNIX Korn shell script to transfer imagery for one sector, create the multispectral AREA file for all four sectors, and generate the composite SST AREA files.

**Input** No input required

**Design Details** The script performs several functions, including:

1. Change directories to the McIDAS data directory (\$HOME/mcidas/data)
2. Prepare the North Atlantic (NATL) images
- 2a. Remove AREA files AREA0530 AREA0531 AREA0532 AREA0540 AREA0541 AREA0542
- 2b. Save older AREA files for multi-epoch screening
  - Copy AREA0131 to AREA0530
  - Copy AREA0132 to AREA0531
  - Copy AREA0141 to AREA0540
  - Copy AREA0142 to AREA0541
- 2c. Average the visible data for GOES-EAST NH sector to 4-km resolution using avgimi program  
avgimi 1950 951 4
- 2d. Generate multispectral GOES AREA file  
gimiglue AREA0532 AREA0951 AREA0952 AREA0954 AREA0955
3. Prepare the SouthAtlantic (SATL) images
- 3a. Remove AREA files AREA0730 AREA0731 AREA0732 AREA0740 AREA0741 AREA0742
- 3b. Save older AREA files for multi-epoch screening
  - Copy AREA0331 to AREA0730
  - Copy AREA0332 to AREA0731
  - Copy AREA0341 to AREA0740
  - Copy AREA0342 to AREA0741
- 3c. Average the visible data for GOES-EAST SH sector to 4-km resolution using avgimi program  
avgimi 1970 971 4
- 3d. Generate multispectral GOES AREA file  
gimiglue AREA0732 AREA0971 AREA0972 AREA0974 AREA0975
4. Prepare the Sotuh Pacific (SPAC) images
- 4a. Remove AREA files AREA0830 AREA0831 AREA0832 AREA0840 AREA0841 AREA0842
- 4b. Save older AREA files for multi-epoch screening
  - Copy AREA0431 to AREA0830
  - Copy AREA0432 to AREA0831
  - Copy AREA0441 to AREA0840
  - Copy AREA0442 to AREA0841
- 4c. Average the visible data for GOES-WEST SH sector to 4-km resolution using avgimi program  
avgimi 1980 981 4
- 4d. Generate multispectral GOES 4-band AREA file  
gimiglue AREA0832 AREA0981 AREA0982 AREA0984 AREA0985
5. Prepare the North Paciic (NPAC) images
- 5a. Remove AREA files AREA0630 AREA0631 AREA0632 AREA0640 AREA0641 AREA0642
- 5b. Save older AREA files for multi-epoch screening
  - Copy AREA0231 to AREA0630
  - Copy AREA0232 to AREA0631
  - Copy AREA0241 to AREA0640
  - Copy AREA0242 to AREA0641

- 5c. Acquire the data for the GOES WEST Northern Hemisphere sector
  - Script calls core McIDAS command imgdel to delete areas 096\*
  - Script calls core McIDAS command imgcopy to copy data from GOES server to local server in McIDAS areas 1960 and 962-965.
- 5d. Average the visible data for GOES-WEST NH sector to 4-km resolution using avgimi program
  - avgimi 1960 961 4
- 5e. Generate multispectral GOES 4-band AREA file
  - gimiglue AREA0632 AREA0961 AREA0962 AREA0964 AREA0965
- 6. Generate GOES SST product as AREA files
  - goes\_sst.k FIRST\_IMAGE=531
- 7. Generate SST AREA composite files (either hourly or 3-hourly depending on time of day)
  - merge.k

**Keywords** None

### 5.1.10 PROCGSST\_HH45.KSH

**Description** A UNIX Korn shell script to transfer imagery for one sector, create the multispectral AREA file for all four sectors, and generate the composite SST AREA files.

**Input** No input required

**Design Details** The script performs several functions, including:

- 1. Change directories to the McIDAS data directory (\$HOME/mcidas/data).
- 2. Prepare the North Atlantic (NATL) images
- 2a. Remove AREA files AREA0130 AREA0131 AREA0132 AREA0140 AREA0141 AREA0142
- 2b. Save older AREA files for multi-epoch screening
  - Copy AREA0531 to AREA0130
  - Copy AREA0532 to AREA0131
  - Copy AREA0541 to AREA0140
  - Copy AREA0542 to AREA0141
- 2c. Average the visible data for GOES-EAST NH sector to 4-km resolution using avgimi program
  - avgimi 1910 911 4
- 2d. Generate multispectral GOES 4-band AREA file
  - gimiglue AREA0132 AREA0911 AREA0912 AREA0914 AREA0915
- 3. Prepare the South Atlantic (SATL) images
- 3a. Remove AREA files AREA0330 AREA0331 AREA0332 AREA0340 AREA0341 AREA0342
- 3b. Save older AREA files for multi-epoch screening
  - Copy AREA0731 to AREA0330
  - Copy AREA0732 to AREA0331
  - Copy AREA0741 to AREA0340
  - Copy AREA0742 to AREA0341

- 3c. Average the visible data for GOES-EAST SH sector to 4-km resolution using avgimi program  
avgimi 1930 931 4
- 3d. Generate multispectral GOES 4-band AREA file  
gimiglue AREA0332 AREA0931 AREA0932 AREA0934 AREA0935
- 4. Prepare the South Pacific (SPAC) images
  - 4a. Remove AREA files AREA0430 AREA0431 AREA0432 AREA0440 AREA0441 AREA0442
  - 4b. Save older AREA files for multi-epoch screening
    - Copy AREA0831 to AREA0430
    - Copy AREA0832 to AREA0431
    - Copy AREA0841 to AREA0440
    - Copy AREA0842 to AREA0441
- 4c. Average the visible data for GOES-WEST SH sector to 4-km resolution using avgimi program  
avgimi 1940 941 4
- 4d. Generate multispectral GOES 4-band AREA file  
gimiglue AREA0432 AREA0941 AREA0942 AREA0944 AREA0945
- 5. Prepare the North Pacific (NPAC) images
  - 5a. Remove AREA files AREA0230 AREA0231 AREA0232 AREA0240 AREA0241 AREA0242
  - 5b. Save older AREA files for multi-epoch screening
    - Copy AREA0631 to AREA0230
    - Copy AREA0632 to AREA0231
    - Copy AREA0641 to AREA0240
    - Copy AREA0642 to AREA0241
- 5c. Acquire the data for the GOES WEST Northern Hemisphere sector
  - Script calls core McIDAS command imgdel to delete areas 092\*
  - Script calls core McIDAS command imgcopy to copy data from GOES server to local server in McIDAS areas 1920 and 922-925.
- 5d. Average the visible data for GOES-WEST NH sector to 4-km resolution using avgimi program  
avgimi 1920 921 4
- 5e. Generate multispectral GOES 4-band AREA file  
gimiglue AREA0232 AREA0921 AREA0922 AREA0924 AREA0925
- 6. Generate GOES SST product as AREA files  
goes\_sst.k FIRST\_IMAGE=131
- 7. Generate buoy matchups  
buoy.k DATE=\$date HOUR=\$hour

**Keywords** None

## **5.2 Main Programs**

### **5.2.1 AVGIMI.PGM (Spatial Reduction Program)**

#### **Description:**

The program, AVGIMI.PGM, is designed to take the GOES visible channel at 1km resolution, and output the GOES visible channel at 4km resolution. The process is to average 16 pixels (i.e. 4 x 4 area) into 1 pixel. The file naming convention of the dataset names are:

#### **Sample Input Files:**

AREA1950 - A GOES 1km visible image

#### **Sample Output Files:**

AREA0951 - A GOES 4km visible image

#### **Command Line:**

avgimi.k infile outfile lres eres

#### **Parameters:**

infile = The input AREA file containing 1km resolution GOES visible data

outfile = The output AREA file containing 4km resolution GOES visible data

lres = The number indicating resolution reduction for lines (default = 1)

eres = The number indicating resolution reduction for elements (default = lres)

#### **Keyword:**

BAND = Band number from multi-banded areas (def = lowest numbered band)

### **5.2.2 BUOY.PGM (Co-locate program)**

#### **Description:**

The program, BUOY.PGM, is designed to co-locate buoy report with GOES measurements. Temporal co-location is done when the time difference among GOES, buoy and the nominal hour is less than 30 minutes. Spatial co-location is done with standard McIDAS navigation. The output file combines the results for each hour.

**Input Files:**

AREA0140-0142, 0240-0242, 0340-0342, 0440-0442 -- For SST files  
BUOYSST.HH -- For buoy data, where HH is the two digit representation of hour (0-23)

**Output Files:**

yyyy\_ddd.hh

where,

yyyy = The four digit representation of the year

ddd = The julian date with a range from 1-365 in a non-leap year, and 1-366 in a leap-year

hh = The two digit representation of the hour, which has a range from 00 - 23.

**Command Line:**

goes\_sst. FIRST\_IMAGE=0531

**Parameters:**

FIRST\_IMAGE =The GOES AREA file used to generate SST AREA files.

**5.2.3 GIMIGLUE.C (Band Merging Program)****Description:**

The program, **GIMIGLUE.C**, reads in the single-band imager area files for a GOES channels 1,2,4,5 and creates a multispectral imager area file. The imager area file serves as the input to the SST product generation program.

**Input Files:**

AREATTCC (where TT=time of day, CC=GOES channel number)

**Output Files:**

AREARE32 (where RE=Sector number (1-4))

**Command Line:**

.GIMIGLUE AREA AREA1 AREA2 AREA3 AREA4

**Parameters:**

AREA = Output Area  
AREA1 = First input area file number (ie. Band 1). This area number defines the characteristics of the output area (ie., number of lines, number elements, resolution, etc.)  
AREA2 = Second input area file number (ie., Band 2).  
AREA4 = Fourth input area file number (ie., Band 4).  
AREA5 = Fifth input area file number (ie., Band 5).

**Keywords:**

BUG = Debug flag (Def=0)  
> 0 for more debug information

### **5.2.4 GOES\_SST.PGM (Main SST Processing Program)**

**Description:**

The main program, GOES\_SST.PGM, reads GOES multispectral AREA files, guess SST, land mask, and solar zenith angle files and outputs a SST file for each sector for each half-hour of the day. The input multi frequency imagery files differ for each half of the hour. They are:

**Input Files:**

AREA0132  
AREA0232  
AREA0332  
AREA0432  
AREA0532  
AREA0632  
AREA0732  
AREA0832

The output files differ likewise for each half of the hour. They are:

**Output Files:**

AREA0142 -- GOES SST Northern Hemisphere sector (Top of the hour)  
AREA0242 -- GOES SST Southern Hemisphere sector (Top of the hour)  
AREA0342 -- GOES SST Northern Hemisphere sector (Top of the hour)  
AREA0442 -- GOES SST Southern Hemisphere sector (Top of the hour)  
AREA0542 -- GOES SST Northern Hemisphere sector (Bottom of the hour)  
AREA0642 -- GOES SST Southern Hemisphere sector (Bottom of the hour)  
AREA0742 -- GOES SST Northern Hemisphere sector (Bottom of the hour)  
AREA0842 -- GOES SST Southern Hemisphere sector (Bottom of the hour)

**Command Line:**

```
goes_sst. FIRST_IMAGE=0531
```

**Parameters:**

FIRST\_IMAGE =The GOES AREA file to process to generate SST AREA files.

**5.2.5 MERGE.PGM (Hourly and 3-hourly composite SST program)****Description:**

The program, MERGE.PGM, is designed to average 3 SST fields w/in a 1 hour period. The purpose is to fill holes created by clouds. To remove stripping, 2 adjacent lines are averaged. This helps to preserve most spatial features.

**Input Files:**

AREA0N4M (where N=Sector number, M=First,Second or Third Epoch)

**Output Files:**

yyyy\_ddd\_hh FOR HOURLY PRODUCTS (where yyyy=year, ddd=julian date, hh=hour)

yyyy\_ddd\_3N FOR 3-HOURLY PRODUCTS (where N=hour/3)

yyyy\_ddd\_3NS FOR HOURLY COASTWATCH PRODUCTS (where S=One letter CoastWatch Sector Code)

**Command Line:**

```
.merge.k
```

**5.3 Subprogram Descriptions****5.3.1 AVGIMI.PGM Subprogram Descriptions****5.3.1.1 ARAGET (AREA,OFFSET,NBYTE,OUTBUF)**

**Description**            Obtain bytes from a McIDAS AREA file

**Input Parameters**    AREA = McIDAS AREA number  
                          OFFSET = First byte to obtain  
                          NBYTE = Number of bytes to obtain

**Output Parameters**   OUTBUF = Array containing the requested data

**Calls** ARANAM(core), ISAFMT(core), LBI, FBYTEC

**Commons** None

### **5.3.1.2 ARAOPT(AREA,NOPT,COPT,VAL)**

**Description** Sets the options used when a McIDAS AREA is read

**Input Parameters** AREA = Area number to set options  
NOPT = Number of options to be set  
COPT = Name of the options to be set  
VAL = Value to set each option to

**Output Parameters** None

**Calls** EDEST (core), ABORT (core), CLIT (core), DDEST (core)

**Commons** None

### **5.3.1.3 ARAPUT(AREA,OFFSET,NBYTE,IARRAY)**

**Description** Add a buffer to an AREA file

**Input Parameters** AREA = McIDAS AREA number  
OFFSET= First byte to obtain  
NBYTE= Number of bytes to obtain  
IARRAY= Array of data to be written

**Output Parameters** None

**Calls** ARANAM(core),ISAFMT(core),LBO,LWO,EDEST(core)

**Commons** None

### **5.4.1.4 CFI(NUMB)**

**Description** Converts an integer to a character\*12, right justified with leading blanks

**Input Parameters** NUMB= Number of elements to convert from integer to character\*12

**Output Parameters** None

**Calls** None

**Commons** None

### 5.3.1.5 CLSARA(AREA)

**Description** Closes a McIDAS AREA file. After calling this routine, the only area access routine that will work is opnara.

**Input Parameters** AREA = Number of McIDAS Area file to be closed

**Output Parameters** None

**Calls** WRTRKA (core), MOVW (core), CLOSAO

**Commons** None

### 5.3.1.6 HOWBIG(AREA,NLIN,NELE)

**Description** Returns number of lines and elements per line in an AREA file

**Input Parameters** AREA = McIDAS AREA number  
NLIN = Number of lines per line in the AREA file  
NELE = Number of elements per line in the AREA file

**Output Parameters** None

**Calls** SDEST(core), READD(core)

**Commons** None

### 5.3.1.7 OPNARA(AREA)

**Description** Function to collect a list of track groups assigned to an area; place them in a common block

**Input Parameters** AREA = Area file number to open

**Output Parameters** None

**Calls** EDEST(core), ABORT(core), CLIT(core), LIT(core), ARAOPT(core), ARADIR(core), READD(core)

**Commons** None

### **5.3.1.8 READD(ANUM,ENTRY)**

**Description**           Reads directory entry from area/sounding directory ‘datdir’

**Input Parameters**   ANUM= Input area number

**Output Parameters**   ENTRY= Output 64 word directory entry

**Calls**               EDEST (core), ARANAM(core),LWI(core),ISAFMT(core),FAHED(core)

**Commons**           None

### **5.3.1.9 REDARA(AREA,LINE,ELEM,NELE,BAND,IARRAY)**

**Description**           Read line from area returning specified subset of line

**Input Parameters**   AREA = Area number  
LINE = Line to read  
ELEM =Starting element desired. 0 returns the first byte of data  
NELE = Number of elements. Will not return any past line end  
BAND = Spectral band of desired data or ‘ALL’ for all bands

**Output Parameters**   IARRAY = Array to hold returned data

**Calls**               EDEST (core), DDEST(core), ABORT (core), FILPFX, LIT (core), ZEROS (core), ISVCOD (core), WRTRKA, SDEST (core), MOVC (core), MOVH(core), MOVPIX (core), FBYTE2 (core), MOVWRD (core), MPIXEL (core), RDTRKA

**Commons**           None

### **5.3.1.10 WRTARA(AREA,LINE,IARRAY)**

**Description**           Write a line to an AREA file

**Input Parameters**   AREA = Area file number  
LINE = Line in area file to write to  
IARRAY =Source of data, length dependent on area size

**Output Parameters**   None

**Calls**               EDEST (core), ABORT (core), FLPPFX, FLPDAT, WRTRKA, MOVB, RDTRKA, FBYTE2

**Commons**           None

### **5.3.1.11 ZEROS(BUF,BYTES)**

**Description** Places zeros in a byte array

**Input Parameters** BYTES = The number of bytes that are zero filled

**Output Parameters** BUF = Array that is zero filled

**Calls** memset

**Commons** None

### **5.3.1.12 ZEROW(NWORDS,IARR)**

**Description** Places zeros in a specified number of words in an array

**Input Parameters** NWORDS = Number of words to be set to zero

**Output Parameters** IARR = The array to be zeroed

**Calls** None

**Commons** None

## **5.3.2 BUOY.PGM Subprogram Descriptions**

### **5.3.2.1 ARAOPT(AREA,NOPT,COPT,VAL)**

**Description** Sets the options used when a McIDAS AREA is read

**Input Parameters** AREA = Area number to set options  
NOPT = Number of options to be set  
COPT = Name of the options to be set  
VAL = Value to set each option to

**Output Parameters** None

**Calls** EDEST (core), ABORT (core), CLIT (core), DDEST (core)

**Commons** None

### **5.3.2.2 CLSARA(AREA)**

<b>Description</b>	Closes a McIDAS AREA file. After calling this routine, the only area access routine that will work is opnara.
<b>Input Parameters</b>	AREA = Number of McIDAS Area file to be closed
<b>Output Parameters</b>	None
<b>Calls</b>	WRTRKA (core), MOVW (core), CLOSAO
<b>Commons</b>	None

### **5.3.2.3 NO\_GOES(IMAGE, REF\_SEC, LAT, LON, SAT\_ID, SAT\_REF, CLEAR, ANGLES, BUF)**

<b>Description</b>	Get the GOES measurements and the derived SST.
<b>Input Parameters</b>	IMAGE= Array containing GOES image REF_SEC=Reference time in seconds LAT,LON=Buoy location
<b>Output Parameters</b>	SAT_ID= Satellite ID SAT_REF=Satellite time - Reference Time CLEAR= The number of clear pixels ANGLES = A array of size three for the viewing geometry BUF = The two dimensional array (11,7) containing the GOES data and SST
<b>Calls</b>	clsara(core), readd(core), mcdaytimetosec(core), nvset(core), opnara(core), nv1eas(core), redara, araopt, nv1opt(core)
<b>Commons</b>	None

### **5.3.2.4 NV1EAS(ZLAT,ZLON,XLIN,XELE,XDUM)**

<b>Description</b>	To transform earth location to satellite coordinates.
<b>Input Parameters</b>	ZLAT= Input Latitude ZLON= Input Longitude XDUM = Unused Argument
<b>Output Parameters</b>	XLIN = Line of satellite image XELE = Element of satellite image

**Calls** EDEST (core), ARANAM(core),LWI(core),ISAFMT(core),FAHED(core)

**Commons** None

### 5.3.2.5 NV1OPT(IFUNC,XIN,XOUT)

**Description** Perform “optional” navigation calculation (e.g. sun angle)

**Input Parameters** IFUNC=4 character string containing which function to perform (i.e. position of sun, satellite and solar elevations and relative azimuth)  
XIN = Array containing parameters relevant to computing that function.

**Output Parameters** XOUT = Array containing the output

**Calls** EDEST (core), ARANAM(core),LWI(core),ISAFMT(core),FAHED(core)

**Commons** None

### 5.3.2.6 NVSET(COPT,NUM)

**Description** Sets up navigation from a codicil. Return values are:

- 0 = Navigation is set up
- 1 = Input area or frame does not exist
- 2 = Navigation data was not found
- 3 = Option was not correct

**Input Parameters** COPT= Identifies where navigation parameters will be taken from (AREA or Frame)  
NUM = McIDAS AREA number

**Output Parameters** None

**Calls** DDEST (core), READD, ARAGET, GETFRM, FRTONR

**Commons** None

### 5.3.2.7 OPNARA(ANUM,ENTRY)

**Description** Function to collect a list of track groups assigned to an area; place them in a common block

**Input Parameters** AREA = Area file number to open

**Output Parameters** None

**Calls** EDEST(core), ABORT(core), CLIT(core), LIT(core), ARAOPT(core), ARADIR(core), READD(core)

**Commons** None

### 5.3.2.8 READD(ANUM,ENTRY)

**Description** Reads directory entry from area/sounding directory ‘datdir’

**Input Parameters** ANUM= Input area number

**Output Parameters** ENTRY= Output 64 word directory entry

**Calls** EDEST (core), ARANAM(core),LWI(core),ISAFMT(core),FAHED(core)

**Commons** None

### 5.3.2.9 REDARA(AREA,LINE,ELEM,NELE,BAND,IARRAY)

**Description** Read line from area returning specified subset of line

**Input Parameters** AREA = Area number

LINE = Line to read

ELEM =Starting element desired. 0 returns the first byte of data

NELE = Number of elements. Will not return any past line end

BAND = Spectral band of desired data or ‘ALL’ for all bands

**Output Parameters** IARRAY = Array to hold returned data

**Calls** EDEST (core), DDEST(core), ABORT (core), FILPFX, LIT (core), ZEROS (core), ISVCOD (core), WRTRKA, SDEST (core), MOVC (core), MOVH(core), MOVPIX (core), FBYTE2 (core), MOVWRD (core), MPIXEL (core), RDTRKA

**Commons** None

## 5.3.3 GOES\_SST.PGM Subprogram Descriptions

### 5.3.3.1 ARAOPT(AREA,NOPT,COPT,VAL)

**Description** Sets the options used when a McIDAS AREA is read

<b>Input Parameters</b>	AREA = Area number to set options NOPT = Number of options to be set COPT = Name of the options to be set VAL = Value to set each option to
<b>Output Parameters</b>	None
<b>Calls</b>	EDEST (core), ABORT (core), CLIT (core), DDEST (core)
<b>Commons</b>	None

### **5.3.3.2 CLSARA(AREA)**

<b>Description</b>	Closes a McIDAS AREA file. After calling this routine, the only area access routine that will work is opnara.
<b>Input Parameters</b>	AREA = Number of McIDAS Area file to be closed
<b>Output Parameters</b>	None
<b>Calls</b>	WRTRKA (core), MOVW (core), CLOSAO
<b>Commons</b>	None

### **5.3.3.3 GOES\_SST(IMAGE)**

<b>Description</b>	Processes the GOES Images
<b>Input Parameters</b>	IMAGE= Area file of image to be navigated
<b>Output Parameters</b>	None
<b>Calls</b>	GETTIM(core),OPNARA(core),ARAOPT(core),READ_AREA,CLSARA(core),STATS,WRTARA,READD(core)
<b>Commons</b>	None

### **5.3.3.4 NAV\_PIX(IMAGE,TWILIGHT,LINE,ELEM,SCENE,SAT\_Z,SUN\_Z)**

<b>Description</b>	Navigate the SST image
<b>Input Parameters</b>	IMAGE= Area file of image to be navigated TWILIGHT=Solar zenith for dawn/evening LINE,ELEM= Pixel image coordinates

**Output Parameters** SCENE= Scene Type (0=space,1=land,2=ocean coast,3=ephemeral water,4=shallow inland water,5=shallow ocean,6=deep inland water,7=deep ocean)  
SAT\_Z=Satellite zenith ((sec(sat\_z)-1)\*1000)  
SUN\_Z= Solar zenith angle (0-180)

**Calls** READD,(core), NVSET(core),NV1opt,SOLARP,NAVLINE,getcoord(core)

**Commons** None

### 5.3.3.5 OPNARA(AREA)

**Description** Function to collect a list of track groups assigned to an area; place them in a common block

**Input Parameters** AREA = Area file number to open

**Output Parameters** None

**Calls** EDEST(core), ABORT(core), CLIT(core), LIT(core), ARAOPT(core), ARADIR(core), READD(core)

**Commons** None

### 5.3.3.6

**READ\_AREA(AREA,LINE,B\_ELEM,N\_ELEM,BAND,BUF,VAL\_LO,VAL\_HI,MIS\_LO,MIS\_HI)**

**Description** Read a line from an area file with quality control.

**Input Parameters** X= 2-D array  
Y=2-D array  
N= The first dimension of the arrays  
JS,JE,IS,IE= The bounds

**Output Parameters** MEAN= The mean  
VARI= The variance (If vari=0, then stats unavailable)

**Calls** None

**Commons** None

### 5.3.3.7 REDARA(AREA,LINE,ELEM,NELE,BAND,IARRAY)

**Description** Read line from area returning specified subset of line

<b>Input Parameters</b>	AREA = Area number LINE = Line to read ELEM = Starting element desired. 0 returns the first byte of data NELE = Number of elements. Will not return any past line end BAND = Spectral band of desired data or 'ALL' for all bands
<b>Output Parameters</b>	IARRAY = Array to hold returned data
<b>Calls</b>	EDEST (core), DDEST(core), ABORT (core), FILPFX, LIT (core), ZEROS (core), ISVCOD (core), WRTRKA, SDEST (core), MOVC (core), MOVH(core), MOVPIX (core), FBYTE2 (core), MOVWRD (core), MPIXEL (core), RDTRKA
<b>Commons</b>	None

### **5.3.3.8 STATS(X,Y,N,JS,JE,IS,IE,MEAN,VARI)**

**Description** Find the basic statistics of the difference between two 2-D arrays.

<b>Input Parameters</b>	X= 2-D array Y=2-D array N= The first dimension of the arrays JS,JE,IS,IE= The bounds
-------------------------	--

<b>Output Parameters</b>	MEAN= The mean VARI = The variance (If vari=0, then stats unavailable)
--------------------------	---

<b>Calls</b>	None
--------------	------

<b>Commons</b>	None
----------------	------

### **5.3.3.9 WRTARA(AREA,LINE,IARRAY)**

**Description** Write a line to an AREA file

<b>Input Parameters</b>	AREA = Area file number LINE = Line in area file to write to IARRAY =Source of data, length dependent on area size
-------------------------	--

<b>Output Parameters</b>	None
--------------------------	------

<b>Calls</b>	EDEST (core), ABORT (core), FLPPFX, FLPDAT, WRTRKA, MOVB, RDTRKA, FBYTE2
--------------	--

<b>Commons</b>	None
----------------	------

### **5.3.4 MERGE.PGM Subprogram Descriptions**

#### **5.3.4.1 ARAOPT(AREA,NOPT,COPT,VAL)**

**Description** Sets the options used when a McIDAS AREA is read

**Input Parameters** AREA = Area number to set options  
NOPT = Number of options to be set  
COPT = Name of the options to be set  
VAL = Value to set each option to

**Output Parameters** None

**Calls** EDEST (core), ABORT (core), CLIT (core), DDEST (core)

**Commons** None

#### **5.3.4.2 CLSARA(AREA)**

**Description** Closes a McIDAS AREA file. After calling this routine, the only area access routine that will work is opnara.

**Input Parameters** AREA = Number of McIDAS Area file to be closed

**Output Parameters** None

**Calls** WRTRKA (core), MOVW (core), CLOSAO

**Commons** None

#### **5.3.4.3 NV1EAS(ZLAT,ZLON,XLIN,XELE,XDUM)**

**Description** To transform earth location to satellite coordinates.

**Input Parameters** ZLAT= Input Latitude  
ZLON= Input Longitude  
XDUM= Unused argument

**Output Parameters** XLIN = Line of satellite image  
XELE = Element of satellite image

**Calls** EDEST (core), ARANAM(core), LWI(core), ISAFMT(core), FAHED(core)

**Commons** None

#### **5.3.4.4 NVSET(COPT,NUM)**

<b>Description</b>	Sets up navigation from a codicil. Return values are:  0 = Navigation is set up -1 = Input area or frame does not exist -2 = Navigation data was not found -3 = Option was not correct
<b>Input Parameters</b>	COPT= Identifies where navigation parameters will be taken from (AREA or Frame) NUM = McIDAS AREA number
<b>Output Parameters</b>	None
<b>Calls</b>	DDEST (core), READD, ARAGET, GETFRM, FRTONR
<b>Commons</b>	None

#### **5.3.4.5 OPNARA(AREA)**

<b>Description</b>	Function to collect a list of track groups assigned to an area; place them in a common block
<b>Input Parameters</b>	AREA = Area file number to open
<b>Output Parameters</b>	None
<b>Calls</b>	EDEST(core), ABORT(core), CLIT(core), LIT(core), ARAOPT(core), ARADIR(core), READD(core)
<b>Commons</b>	None

#### **5.3.4.6 READD(ANUM,ENTRY)**

<b>Description</b>	Reads directory entry from area/sounding directory ‘datdir’
<b>Input Parameters</b>	ANUM= Input area number
<b>Output Parameters</b>	ENTRY= Output 64 word directory entry
<b>Calls</b>	EDEST (core), ARANAM(core), LWI(core), ISAFMT(core), FAHED(core)
<b>Commons</b>	None

### **5.3.4.7 REDARA(AREA,LINE,ELEM,NELE,BAND,IARRAY)**

<b>Description</b>	Read line from area returning specified subset of line
<b>Input Parameters</b>	AREA = Area number LINE = Line to read ELEM = Starting element desired. 0 returns the first byte of data NELE = Number of elements. Will not return any past line end BAND = Spectral band of desired data or 'ALL' for all bands
<b>Output Parameters</b>	IARRAY = Array to hold returned data
<b>Calls</b>	EDEST (core), DDEST(core), ABORT (core), FILPFX, LIT (core), ZEROS (core), ISVCOD (core), WRTRKA, SDEST (core), MOVC (core), MOVH(core), MOVPIX (core), FBYTE2 (core), MOVWRD (core), MPIXEL (core), RDTRKA
<b>Commons</b>	None

## **6. Database Specification**

### **6.1 Input File Specifications**

#### **6.1.1 Calibration Coefficient files**

This file, REG\_COEFF.ASC, contains the coefficients that are used to take satellite radiances and convert to SST. The four lines correspond to instrument id for GOES-8, five retrieval constants for daytime, five for nighttime, id for GOES-10, and five retrieval constants for daytime, five for nighttime.

### **6.2 Output File Specifications**

#### **6.2.1 GETDATA\_HH15.LOG**

This log file contains information about transfer of the GOES AREA files for 3 of the 4 sector regions for the top of the hour. The following is a sample output:

```
Thu Dec 7 08:05:04 UTC 2000 start getdata_hh15.ksh
GOES-E NHem image between 7:35 and 7:55 on date 2000342
imgdel.k: done
imgdel.k: done
imgcopy.k GER/GENHEM01V LD/A.1950 PLACE=ULEFT LINELE=2805 9049 I BAND=ALL DOC=YES
SIZE=7308 13840 DAY=2000342 TIME=7:35 7:55
Beginning Image Data transfer, bytes= 203984384
Transferring AREA data outbound, bytes= 203984544
IMGCOPY: GER/GENHEM01V.4 copied to LD/A.1950
imgcopy.k: done
Beginning Image Data transfer, bytes= 13070192
Transferring AREA data outbound, bytes= 13070352
IMGCOPY: GER/GENHEM04I2.25 copied to LD/A.952
imgcopy.k: done
Beginning Image Data transfer, bytes= 13070192
Transferring AREA data outbound, bytes= 13070352
IMGCOPY: GER/GENHEM04I3.25 copied to LD/A.953
imgcopy.k: done
Beginning Image Data transfer, bytes= 13070192
Transferring AREA data outbound, bytes= 13070352
IMGCOPY: GER/GENHEM04I4.25 copied to LD/A.954
imgcopy.k: done
Beginning Image Data transfer, bytes= 13070192
Transferring AREA data outbound, bytes= 13070352
IMGCOPY: GER/GENHEM04I5.25 copied to LD/A.955
imgcopy.k: done
Got GOES-E NHem at Thu Dec 7 08:08:20 UTC 2000
GOES-E SHem image between 7:35 and 7:45 on date 2000342
imgdel.k: done
imgdel.k: done
```

Beginning Image Data transfer, bytes= 4387664  
Transferring AREA data outbound, bytes= 4387824  
IMGCOPY: GER/GESHEM04I2.43 copied to LD/A.972  
imgcopy.k: done  
Beginning Image Data transfer, bytes= 4387664  
Transferring AREA data outbound, bytes= 4387824  
IMGCOPY: GER/GESHEM04I3.43 copied to LD/A.973  
imgcopy.k: done  
Beginning Image Data transfer, bytes= 4387664  
Transferring AREA data outbound, bytes= 4387824  
IMGCOPY: GER/GESHEM04I4.43 copied to LD/A.974  
imgcopy.k: done  
Beginning Image Data transfer, bytes= 4387664  
Transferring AREA data outbound, bytes= 4387824  
IMGCOPY: GER/GESHEM04I5.43 copied to LD/A.975  
imgcopy.k: done  
Got GOES-E SHem at Thu Dec 7 08:09:36 UTC 2000  
GOES-W SHem image between 7:35 and 7:55 on date 2000342  
imgdel.k: done  
imgdel.k: done  
imgcopy.k GWR/GWSHEM01V LD/A.1980 PLACE=ULEFT LINELE=7885 7845 I BAND=ALL DOC=YES  
SIZE=4252 13600 DAY=2000342 TIME=7:35 7:55  
Beginning Image Data transfer, bytes= 94908128  
Transferring AREA data outbound, bytes= 116644512  
IMGCOPY: GWR/GWSHEM01V.12 copied to LD/A.1980  
imgcopy.k: done  
Beginning Image Data transfer, bytes= 6126368  
Transferring AREA data outbound, bytes= 7478664  
IMGCOPY: GWR/GWSHEM04I2.13 copied to LD/A.982  
imgcopy.k: done  
Beginning Image Data transfer, bytes= 6126368  
Transferring AREA data outbound, bytes= 7478664  
IMGCOPY: GWR/GWSHEM04I3.13 copied to LD/A.983  
imgcopy.k: done  
Beginning Image Data transfer, bytes= 6126368  
Transferring AREA data outbound, bytes= 7478664  
IMGCOPY: GWR/GWSHEM04I4.13 copied to LD/A.984  
imgcopy.k: done  
Beginning Image Data transfer, bytes= 6126368  
Transferring AREA data outbound, bytes= 7478664  
IMGCOPY: GWR/GWSHEM04I5.13 copied to LD/A.985  
imgcopy.k: done  
Got GOES-W SHem at Thu Dec 7 08:11:46 UTC 2000  
Thu Dec 7 08:11:47 UTC 2000 done copying data  
Thu Dec 7 08:11:47 UTC 2000 end getdata\_hh15.ksh

## **6.2.2 PROCGSST\_HH15.LOG**

This log file contains information about transfer of the GOES AREA files for the 4th of the sector regions and information regarding the processing to make the half-hourly, hourly and 3-hourly SST products for the top of the hour. The following is a sample output:

```
Thu Dec 7 08:29:01 UTC 2000 start procsst_hh15.ksh
GOES-E NHem image between 7:35 and 7:55 on date 2000342
Timetest: start avgimi at Thu Dec 7 08:29:39 UTC 2000
Begin averaging
Line reduction factor:      4
Element reduction factor:   4
avgimi.k: DONE 951
 0   0   0
 0   0   0
 0   0   0
 0   0   0
 0   8   10   0
before first araput...LEN is      2560
before secnd araget
len 512 2816
before secnd araput
Timetest: end avgimi/start gimiglue at Thu Dec 7 08:30:20 UTC 2000
number of arguements 7
gimiglue
AREA0532 AREA0951 AREA0952 AREA0953 AREA0954 AREA0955
opnara fd 3
output area bounds
top left line 2805 element 9049
bot right line 10109 element 22885
band locations: 1 2 3 4 5
area0 channel map 0000001f
prefix documentation length mismatch file 2 length 0 should be 228
prefix documentation length mismatch file 3 length 0 should be 228
prefix documentation length mismatch file 4 length 0 should be 228
in gcombine
done with gimiglue
Timetest: end gimiglue at Thu Dec 7 08:30:48 UTC 2000
UX:cp: ERROR: AREA0341 - No such file or directory
GOES-E SHem image between 7:35 and 7:44 on date 2000342
Timetest: start avgimi at Thu Dec 7 08:30:55 UTC 2000
Begin averaging
Line reduction factor:      4
Element reduction factor:   4
```

```

avgimi.k: DONE 971
 0   0   0
 0   0   0
 0   0   0
 0   0   0
 0   8   10   0
before first araput...LEN is      2560
before secnd araget
len  512 2816
before secnd araput
Timetest: end avgimi/start gimiglue at Thu Dec 7 08:31:10 UTC 2000
number of arguements  7
gimiglue
AREA0732 AREA0971 AREA0972 AREA0973 AREA0974 AREA0975
opnara fd 3
output area bounds
top left line 10037 element 9049
bot right line 12485 element 22885
band locations: 1 2 3 4 5
area0 channel map 0000001f
prefix documentation length mismatch file 2 length 0 should be 228
prefix documentation length mismatch file 3 length 0 should be 228
prefix documentation length mismatch file 4 length 0 should be 228
in gcombine
done with gimiglue
Timetest: end gimiglue at Thu Dec 7 08:31:18 UTC 2000
GOES-W SHem image between 7:35 and 7:55 on date 2000342
Timetest: start avgimi at Thu Dec 7 08:31:42 UTC 2000
Begin averaging
Line reduction factor:      4
Element reduction factor:  4
avgimi.k: DONE 981
 0   0   0
 0   0   0
 0   0   0
 0   0   0
 0   8   10   0
before first araput...LEN is      2560
before secnd araget
len  512 2816
before secnd araput
Timetest: end avgimi/start gimiglue at Thu Dec 7 08:32:06 UTC 2000
number of arguements  7

```

### **6.2.3 GETDATA\_HH45.LOG**

This log file contains information about transfer of the GOES AREA files for 3 of the 4 sector regions for the bottom of the hour. The following is a sample output:

```
GOES-E NHem image between 9:05 and 9:25 on date 2000342
imgdel.k: done
imgdel.k: done
imgcopy.k GER/GENHEM01V LD/A.1910 PLACE=ULEFT LINELE=2805 9049 I BAND=ALL DOC=YES
SIZE=7308 13840 DAY=2000342 TIME=9:05 9:25
Beginning Image Data transfer, bytes= 203984384
Transferring AREA data outbound, bytes= 203984544
IMGCOPY: GER/GENHEM01V.7 copied to LD/A.1910
imgcopy.k: done
Beginning Image Data transfer, bytes= 13070192
Transferring AREA data outbound, bytes= 13070352
IMGCOPY: GER/GENHEM04I2.28 copied to LD/A.912
imgcopy.k: done
Beginning Image Data transfer, bytes= 13070192
Transferring AREA data outbound, bytes= 13070352
IMGCOPY: GER/GENHEM04I3.28 copied to LD/A.913
imgcopy.k: done
Beginning Image Data transfer, bytes= 13070192
Transferring AREA data outbound, bytes= 13070352
IMGCOPY: GER/GENHEM04I4.28 copied to LD/A.914
imgcopy.k: done
Beginning Image Data transfer, bytes= 13070192
Transferring AREA data outbound, bytes= 13070352
IMGCOPY: GER/GENHEM04I5.28 copied to LD/A.915
imgcopy.k: done
Got GOES-E NHem at Thu Dec 7 09:43:45 UTC 2000
GOES-E SHem image between :40 and :50 on date
imgdel.k: done
imgdel.k: done
imgcopy.k GER/GESHEM01V LD/A.1930 PLACE=ULEFT LINELE=10037 9049 I BAND=ALL DOC=YES
SIZE_GESV DAY=2000342 TIME=8:40 8:50
Beginning Image Data transfer, bytes= 729248
Transferring AREA data outbound, bytes= 729408
IMGCOPY: GER/GESHEM01V.6 copied to LD/A.1930
imgcopy.k: done
Beginning Image Data transfer, bytes= 729248
```

Transferring AREA data outbound, bytes= 729408  
IMGCOPY: GER/GESHEM04I2.46 copied to LD/A.932  
imgcopy.k: done  
Beginning Image Data transfer, bytes= 729248  
Transferring AREA data outbound, bytes= 729408  
IMGCOPY: GER/GESHEM04I3.46 copied to LD/A.933  
imgcopy.k: done  
Beginning Image Data transfer, bytes= 729248  
Transferring AREA data outbound, bytes= 729408  
IMGCOPY: GER/GESHEM04I4.46 copied to LD/A.934  
imgcopy.k: done  
Beginning Image Data transfer, bytes= 729248  
Transferring AREA data outbound, bytes= 729408  
IMGCOPY: GER/GESHEM04I5.46 copied to LD/A.935  
imgcopy.k: done  
Got GOES-E SHem at Thu Dec 7 09:44:08 UTC 2000  
GOES-W SHem image between 9:00 and 9:30 on date 2000342  
imgdel.k: done  
imgdel.k: done  
imgcopy.k GWR/GWSHEM'01V' LD/A.1940 PLACE=ULEFT LINELE=7885 7845 I BAND=ALL DOC=YES  
SIZE=4252 13600 DAY=2000342 TIME=9:00 9:30  
imgcopy.k: No images satisfy the selection criteria  
imgcopy.k: done  
Beginning Image Data transfer, bytes= 6103328  
Transferring AREA data outbound, bytes= 7478664  
IMGCOPY: GWR/GWSHEM04I2.16 copied to LD/A.942  
imgcopy.k: done  
Beginning Image Data transfer, bytes= 6103328  
Transferring AREA data outbound, bytes= 7478664  
IMGCOPY: GWR/GWSHEM04I3.16 copied to LD/A.943  
imgcopy.k: done  
Beginning Image Data transfer, bytes= 6103328  
Transferring AREA data outbound, bytes= 7478664  
IMGCOPY: GWR/GWSHEM04I4.16 copied to LD/A.944  
imgcopy.k: done  
Beginning Image Data transfer, bytes= 6103328  
Transferring AREA data outbound, bytes= 7478664  
IMGCOPY: GWR/GWSHEM04I5.16 copied to LD/A.945  
imgcopy.k: done  
Got GOES-W SHem at Thu Dec 7 09:44:44 UTC 2000  
Thu Dec 7 09:44:44 UTC 2000 done copying data  
Thu Dec 7 09:44:44 UTC 2000 end getdata\_hh45.ksh

## **6.2.4 PROCGSST\_HH45.LOG**

This log file contains information about transfer of the GOES AREA files for the 4th of the sector regions and information regarding the processing to make the half-hourly, hourly SST products and the buoy matchups for the bottom of the hour. The following is a sample output:

```
Thu Dec 7 10:59:00 UTC 2000 start procsst_hh45.ksh
GOES-E NHem image between 10:05 and 10:25 on date 2000342
Timetest: start avgimi at Thu Dec 7 10:59:38 UTC 2000
Begin averaging
Line reduction factor:      4
Element reduction factor:   4
avgimi.k: DONE 911
  0   0   0
  0   0   0
  0   0   0
  0   0   0
  0   8   10   0
before first araput...LEN is      2560
before secnd araget
len   512  2816
before secnd araput
Timetest: end avgimi/start gimiglue at Thu Dec 7 11:00:06 UTC 2000
number of arguements   7
gimiglue
AREA0132 AREA0911 AREA0912 AREA0913 AREA0914 AREA0915
opnara fd  3
output area bounds
top left line 2805 element 9049
bot right line 10109 element 22885
band locations: 1 2 3 4 5
area0 channel map 0000001f
prefix documentation length mismatch file 2 length 0 should be 228
prefix documentation length mismatch file 3 length 0 should be 228
prefix documentation length mismatch file 4 length 0 should be 228
in gcombine
done with gimiglue
Timetest: end gimiglue at Thu Dec 7 11:00:32 UTC 2000
GOES-E SHem image between 10:05 and 10:15 on date 2000342
Timetest: start avgimi at Thu Dec 7 11:00:38 UTC 2000
Begin averaging
Line reduction factor:      4
Element reduction factor:   4
```

```

avgimi.k: DONE 931
 0   0   0
 0   0   0
 0   0   0
 0   0   0
 0   8   10   0
before first araput...LEN is      2560
before secnd araget
len  512 2816
before secnd araput
Timetest: end avgimi/start gimiglue at Thu Dec 7 11:00:40 UTC 2000
number of arguements 7
gimiglue
AREA0332 AREA0931 AREA0932 AREA0933 AREA0934 AREA0935
opnara fd 3
output area bounds
top left line 10037 element 9049
bot right line 11953 element 11605
band locations: 1 2 3 4 5
area0 channel map 0000001f
prefix documentation length mismatch file 2 length 0 should be 228
prefix documentation length mismatch file 3 length 0 should be 228
prefix documentation length mismatch file 4 length 0 should be 228
in gcombine
done with gimiglue
Timetest: end gimiglue at Thu Dec 7 11:00:41 UTC 2000
UX:cp: ERROR: AREA0831 - No such file or directory
UX:cp: ERROR: AREA0841 - No such file or directory
UX:cp: ERROR: AREA0842 - No such file or directory
GOES-W SHem image between 10:00 and 10:30 on date 2000342
Timetest: start avgimi at Thu Dec 7 11:00:41 UTC 2000
Begin averaging
Line reduction factor:      4
Element reduction factor: 4
avgimi.k: DONE 941
 0   0   0
 0   0   0
 0   0   0
 0   0   0
 0   8   10   0
before first araput...LEN is      2560
before secnd araget
len  512 2816
before secnd araput
Timetest: end avgimi/start gimiglue at Thu Dec 7 08:32:06 UTC 2000
number of arguements 7

```

## 7. Installation Procedures

### 7.1 Installation of Software and Data

The steps involved in installation are relatively straightforward assuming a tape backup of the original software/data is made. These backups should be made periodically to keep up with any changes which may occur to the software/data. Below is a list of the steps involved in installing the software/data on a new machine.

**Important Note:** A backup tape of current software/data on the ORIGIN 2000 (GP3) must be made beforehand in order to perform this installation.

- A)** Make a tape backup of the original workstation using the “tar” command. The system administrator should know how to execute the “tar” command, as it is standard UNIX. The process would be to:

```
cd $HOME/mcidas  
tar -cvf gsstprogs.tar src lib bin ksh data/
```

The following directories should be backed up for the SST product processing system:

1. \$HOME
2. \$HOME/mcidas
3. \$HOME/mcidas/src
4. \$HOMEmcidas/ksh
5. \$HOME/mcidas/lib
6. \$HOME/mcidas/bin
7. \$HOME/mcidas/data

Selected files for \$HOME/mcidas/data include:

```
reg_coef.*  
link_guess  
AREA0100 ... 0123. AREA0200 ... 0223, AREA0300 ... 0323, AREA0400 ... 0423
```

- B)** Install McIDAS-X on the workstation to be used.
- C)** Create the above directories using the “mkdir” command. The system administrator should know how to do this.
- D)** Using the “tar” command, copy all software and data off of the tape and into the proper directories listed above.

**Note:** The instructions below are meant to serve merely as a **guide** during configuration and may have to be modified slightly to suit the users needs.

- E)** Recompile all of the SST software, which resides in a single directory (\$HOME/mcidas/src) under the goessst user account. The compile script resides in the ~/mcidas/src directory and is named **compile\_gsst**.

The procedure, then, to compile the SST software is simply to run the **compile\_gsst** executable by doing the following:

- 1) Login
- 2) Change directory to the \$HOME/mcidas/src directory (i.e. type cd \$HOME/mcidas/src).
- 3) Compile the SST software by typing: ‘**compile\_gsst**’

- F)** Restore the crontab file to schedule execution of the jobs (cd \$HOME/mcidas/data, crontab crontab.save).
- G)** Revise scripts if archive directory or product server name/directory has changed. Revise \$HOME/.netrc to include product server, buoy data server. Verify that all scripts are executable.
- H)** Have the system administrator ensure that the product server has permission to allow data to be placed on the new machine (ftp access).
- I)** Configure ADDE to access and store the satellite imagery. This involves using the McIDAS DATALOC and DSERVE commands. The DATALOC command is used when accessing a remote server while the DSERVE command sets up the local server. The following commands need to be entered at the McIDAS command line:
- 1) DATALOC ADD GER 140.90.105.41
  - 2) DATALOC ADD GWR 140.90.105.51
  - 3) DSERVE ADD LD/A AREA 1 9999 “ALL LOCAL AREAS”

## **7.2 Installation Changes Concerning System/Configuration Upgrades**

Numerous different types of system configuration changes or system upgrades may result in changes in installation procedures or other system level changes. This section discusses briefly some of the more important changes.

### **7.2.1 McIDAS-X Upgrades**

The system administrator should be fully aware of what needs to be done for a McIDAS-X upgrade. At the very least the following steps should be completed. Once a new McIDAS upgrade is installed, recompile the mcidas software (see section 7.1 E).

```
cp $HOME/mcidas/lib/libmcidas.a $HOME/lib
```

Now follow the instructions in Section 7.1 for compiling the SST software.

The above will bring in the new libmcidas.a file. The user will then re-compile the SST source code by running **compile\_gsst**.

## **8. Utilities For System Observation And Data Analysis**

This section contains a listing of several programs which can aid a system analyst or operator in looking at the data. The use could be purely for observational purposes or for analytical purposes or for system maintenance (i.e., troubleshooting) purposes. There are many utilities serving diverse purposes.

### **8.1 Utilities For System Observation/Maintenance/Analysis**

This section contains a listing of several programs which can aid a system analyst or operator in looking at the data. The use could be purely for observational purposes or for analytical purposes or for system maintenance (i.e., troubleshooting) purposes. There are many utilities serving diverse purposes.

#### **8.1.1 Utility Listing**

<b>Area Files:</b>	<b>IMGLIST:</b> Lists image directory information
	<b>IMGDISP:</b> Displays image data from a server to a frame

#### **8.1.2 Utility Examples**

The utilities IMPLIST and IMPDISP are all core utilities. Therefore, help is available on them if needed. Below is a list of examples to aid the user in utilizing these programs on files used or generated by the SST processing system.

##### **IMGLIST LD/A.0130 013**

Lists the local multiband imagery files for NATL region.

##### **IMGDISP LD/A.0142 MAG=-4 -8**

Displays the latest SST result for NATL. The SST output imagery is sampled every fourth line and every eighth element.

## **9. References**

McIDAS-X Users Guide. Space Science and Engineering Center. University of Wisconsin-Madison. 1225 West Dayton Street. Madison, WI 53706.

McIDAS-X Learning Guide. Space Science and Engineering Center. University of Wisconsin-Madison. 1225 West Dayton Street. Madison, WI 53706.

McIDAS-X Programmers Guide. Space Science and Engineering Center. University of Wisconsin-Madison. 1225 West Dayton Street. Madison, WI 53706.

DDE Demonstration Manual. Space Science and Engineering Center. University of Wisconsin-Madison. 1225 West Dayton Street. Madison, WI 53706.